

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-85597

(43)公開日 平成11年(1999)3月30日

(51)Int.Cl.⁶

G 06 F 12/00

識別記号

535

F I

G 06 F 12/00

535Z

審査請求 未請求 請求項の数13 O.L (全 13 頁)

(21)出願番号 特願平9-239742

(22)出願日 平成9年(1997)9月4日

(71)出願人 000005234

富士電機株式会社

神奈川県川崎市川崎区田辺新田1番1号

(71)出願人 000237156

株式会社エフ・エフ・シー

東京都日野市富士町1番地

(72)発明者 高野 幸雄

東京都日野市富士町1番地 富士ファコム

制御株式会社内

(72)発明者 小山 新一郎

東京都日野市富士町1番地 富士ファコム

制御株式会社内

(74)代理人 弁理士 大曾 義之

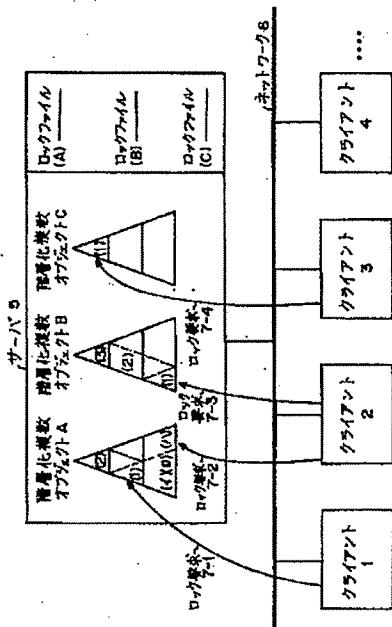
(54)【発明の名称】階層化された複数オブジェクトのロック制御方法

(57)【要約】

【課題】階層化された複数のオブジェクトのロック制御において、あるオブジェクトに対してロック要求があった場合、そのオブジェクトと親子関係にある全てのオブジェクトのみをロックするロック制御を実現する方法を提供する。

【解決手段】階層化された複数のオブジェクトA、B、Cの制御が可能な情報処理システム環境で、クライアント1から1つのオブジェクトA-(1)に対してロック要求がなされた場合に、そのオブジェクトA-(1)がロック中でなければ、そのオブジェクトA-(1)の名称、及び、階層構造の中での階層位置と、そのオブジェクトA-(1)と親子関係にある上位階層のオブジェクトA-(2)の名称を含むロックファイル(A)を生成し、そのロックファイル(A)により、そのオブジェクトA-(1)、そのオブジェクトA-(1)と親子関係にある上位階層の全てのオブジェクトA-(2)、及び、そのオブジェクトA-(1)と親子関係にある下位階層の全てのオブジェクトA-(4)、A-(5)をロックする。

全体構成を示す図



【特許請求の範囲】

【請求項1】階層化された複数のオブジェクトの記憶と制御及びそれらのオブジェクトを用いた処理が可能な情報処理システム環境において、

1つのオブジェクトに対してロック要求がなされた場合に、前記ロック要求のあった前記オブジェクトと、前記オブジェクトが属する階層より下位の全ての階層の前記オブジェクトの全ての子オブジェクトと、前記オブジェクトが属する階層より上位の全ての階層の前記オブジェクトの全ての親オブジェクトのみに対して、ロックを行うことを特徴とする階層化された複数のオブジェクトのロック制御方法。

【請求項2】前記ロック要求のあった前記オブジェクトについて、

少なくとも前記オブジェクトが前記階層化された複数のオブジェクトの階層構造中のどの階層位置に属するかを示す第1の情報と、前記オブジェクトの名称及び前記オブジェクトの全ての親オブジェクトの名称に関する第2の情報を有するロックファイルの生成と削除によって、前記ロック制御を行うことを特徴とする請求項1に記載の階層化された複数のオブジェクトのロック制御方法。

【請求項3】前記ロックファイルは、更に、前記階層化された複数のオブジェクトの階層構造の階層数を示す第3の情報を有することを特徴とする請求項2に記載の階層化された複数のオブジェクトのロック制御方法。

【請求項4】前記第1、第2及び第3の情報は、前記ロックファイルの名称又はレコードに設定されることを特徴とする請求項3に記載の階層化された複数のオブジェクトのロック制御方法。

【請求項5】前記ロックファイルの名称又はレコードは、前記ロック要求のあったオブジェクトの名称と該オブジェクトの全ての親オブジェクトの名称が、最上位の階層のオブジェクトから階層の上位順に特定の記号により区切って配列され、該ロック要求のあったオブジェクトの名称の後に続く前記特定の記号の後方には、前記特定の記号が該ロック要求のあったオブジェクトが属する階層よりも下位にある全ての階層数と同数配列されて成る文字列から成り、

前記第1の情報は、前記ロック要求のあったオブジェクトの名称に続く前記特定の記号の該文字列中における出現順位によって示され、

前記第2の情報は該文字列中の各オブジェクト名称によって示され、

該文字列中において最後に配列されたオブジェクト名称が前記ロック要求のあったオブジェクトの名称を表し、該文字列中におけるその他のオブジェクト名称が前記全ての親オブジェクトの名称を表し、前記第3の情報は、該文字列中の前記特定の記号の総数によって示されることを特徴とする請求項4に記載の階層化された複数のオブジェクトのロック制御方法。

【請求項6】新規のロック要求が発生した場合、該ロック要求のあったオブジェクトの名称を基に、該オブジェクトがロックされていることを示す前記ロックファイルが存在するか否かを検索することにより、上記新規のロック要求のあったオブジェクトが既にロックされているか否かを判定することを特徴とする請求項2、3、4又は5に記載の階層化された複数のオブジェクトのロック制御方法。

【請求項7】階層化された複数のオブジェクトの記憶と制御及びそれらのオブジェクトを用いた処理が可能な情報処理システム環境において、

複数のオブジェクトに対する一括ロック要求がなされた場合に、前記ロック要求のあった各オブジェクト個々に対して、該各オブジェクトが属する階層より下位の全ての階層の該各オブジェクトの全ての子オブジェクトと、該各オブジェクトが属する階層より上位の全ての階層の該各オブジェクトの全ての親オブジェクトのみに対してロックを試み、

1つでもロックできないオブジェクトが存在する時には前記ロック要求のあった全てのオブジェクトに対してロックを行わないことを特徴とする階層化された複数のオブジェクトのロック制御方法。

【請求項8】階層化された複数のオブジェクトの記憶と制御及びそれらのオブジェクトを用いた処理が可能な情報処理システム環境において、

複数のオブジェクトに対する一括ロック要求がなされた場合に、前記ロック要求のあった各オブジェクト個々に対して、該各オブジェクトが属する階層より下位の全ての階層の該各オブジェクトの全ての子オブジェクトと、該各オブジェクトが属する階層より上位の全ての階層の該各オブジェクトの全ての親オブジェクトのみに対してロックを試み、
既にロックされているオブジェクトは除外し、ロック可能なオブジェクトのみをロックすることを特徴とする階層化された複数のオブジェクトのロック制御方法。

【請求項9】複数のオブジェクトに対して、一括してロック要求を行う場合、請求項7記載のロック制御方法又は請求項8記載のロック制御方法のどちらか一方を、前記ロック要求時に指定することを特徴とする階層化された複数のオブジェクトのロック制御方法。

【請求項10】前記ロック要求に対する階層化された複数のオブジェクトのロック制御を、クライアント/サーバ環境においてサーバにより一元的に統括して行うことを特徴とする請求項1から請求項9のいずれか一つに記載の階層化された複数のオブジェクトのロック制御方法。

【請求項11】クライアント側で、複数のオブジェクトに対するロック要求、又は、複数のオブジェクトに対してロックが可能か否かの判定要求が発生した場合には、それらの要求を一括して前記サーバに依頼することを特

徴とする請求項10に記載の階層化された複数のオブジェクトのロック制御方法。

【請求項12】階層化された複数のオブジェクトの記憶と制御及びそれらのオブジェクトを用いた処理が可能な情報処理システム環境で1つのオブジェクトに対してロック要求がなされた場合に、該オブジェクトに対してロックをかけることが可能な装置において、

少なくとも前記オブジェクトが前記階層化された複数のオブジェクトの階層構造中のどの階層位置に属するかを示す第1の情報と、前記オブジェクトの名称及び前記オブジェクトの全ての親オブジェクトの名称に関する第2の情報を有するロックファイルの生成と削除によって、前記ロックをかけるための制御を行うロックファイル生成／削除手段と、

新規のロック要求が発生した場合、該ロック要求のあつたオブジェクトの名称を基に、該オブジェクトがロックされていることを示す前記ロックファイルが存在するか否かを検索することにより、上記新規のロック要求のあつたオブジェクトが既にロックされているか否かを判定する判定手段と、

該判定手段の判定結果に基づいて、前記ロック要求がなされたオブジェクトがロック可能な場合には前記ロックファイル生成／削除手段に、該オブジェクトに対応するロックファイルを生成させる制御手段とから構成されることを特徴とする階層化された複数のオブジェクトのロック制御装置。

【請求項13】階層化された複数のオブジェクトの記憶と制御及びそのオブジェクトを用いた処理が可能な情報処理システム環境で1つのオブジェクトに対してロック要求がなされた場合において、

少なくとも前記オブジェクトが前記階層化された複数のオブジェクトの階層構造中のどの階層位置に属するかを示す第1の情報と、前記オブジェクトの名称及び前記オブジェクトの全ての親オブジェクトの名称に関する第2の情報を有するロックファイルを生成する手順と、

該ロックファイルにより他のロック要求の排他処理を行う手順と、

をコンピュータに実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、情報処理システム環境における階層化された複数のオブジェクトに対するロック（アクセスの排他）制御方法に関する。

【0002】

【従来の技術】従来の情報処理システム環境における階層化された複数のオブジェクトのロック制御方法では、一つのオブジェクトに対して内容の変更を含む第1のアクセスの排他要求（ロック要求：以下、第1ロック要求と表記）による処理中に、そのオブジェクトに対して内

容の変更を含む第2のアクセスの排他要求（以下、第2ロック要求と表記）が有った場合には、その第1ロック要求による処理中は、第2ロック要求について拒否されるロック（排他）制御が知られている。

【0003】又、そのロックの内容としては、その階層化された複数のオブジェクト全体を第1ロック要求によりロックして第2ロック要求以降のロック要求の排他を行うロック制御を行うか、アクセスの排他要求のあつた一つのオブジェクトについてのみ第1ロック要求によりロックして第2ロック要求以降のロック要求の排他制御を行う制御が知られている。

【0004】前者の階層化された複数のオブジェクト全体をロックする排他制御の場合には、第1ロック要求による1つのオブジェクトの処理が終わるまでは第2ロック要求以降はそのオブジェクトと親子関係にないオブジェクトについても処理ができずに、第1ロック要求の処理が終了するのを待つかなかった。

【0005】一方、後者のロック要求のあつたオブジェクトのみをロックする排他制御の場合には、第1ロック要求による処理中でも、処理中のオブジェクト以外のオブジェクトについては第2ロック要求以降での処理は可能であったが、第1のロック要求による一つのオブジェクトの処理中に、第2ロック要求以降の処理でその第1のロック要求で処理中のオブジェクトと親子関係にある別オブジェクトの処理を行ってしまうと、オブジェクト同士の親子内容に矛盾が生じてしまうことがあった。

【0006】尚、本発明の明細書における親子関係とは、当該オブジェクトに対して親、祖父母、曾祖父母や子、孫、曾孫等の関係にあることを意味するものとする。例えば、親オブジェクトという場合は、当該オブジェクトに対して、インヘリタンスにより、親、祖父母、曾祖父母等の関係にある、当該オブジェクトが属する階層よりも上位の階層に属している全てのオブジェクトのことを意味する。

【0007】又、子オブジェクトという場合は、当該オブジェクトに対して、インヘリタンスにより、子、孫、曾孫等の関係にある、当該オブジェクトが属する階層よりも下位の階層に属している全てのオブジェクトのことを意味する。従って、本発明の明細書においては、親子関係、親オブジェクト、子オブジェクト等の表記は全て、上記した広義の親子関係、親オブジェクト、子オブジェクトを示すものである。

【0008】

【発明が解決しようとする課題】従来は、階層化された複数のオブジェクトの中のあるオブジェクトについて、そのオブジェクトの階層中の階層位置を示す方法や、そのオブジェクトと親子関係にある他のオブジェクトを示す方法が無いため、そのオブジェクトの変更中においては、そのオブジェクトと親子関係にあるオブジェクトを判定して、そのオブジェクト及びそのオブジェクトと親

子関係にあるオブジェクトのみに対してロックを掛けようとしても、そのオブジェクトと親子関係にあるオブジェクトを判定する方法が無いためできなかった。

【0009】従って、ロック要求のあったオブジェクト及びそのオブジェクトと親子関係にあるオブジェクトを含めてロックする有効かつ簡潔な制御技術は確立されていないので、階層化された複数のオブジェクトの中の任意のオブジェクトに対してロック要求が有った場合は、オブジェクトデータの利用効率を犠牲にして階層化された複数のオブジェクト全体をロックするか、オブジェクト同士の親子関係が崩れるリスクを負うことを承知でロック要求の有ったオブジェクトのみをロックするしか方法は無かった。

【0010】又、クライアント/サーバ環境においても、ロック要求のあったオブジェクトとそのオブジェクトと親子関係にあるオブジェクトのみをロックする有効かつ簡潔な制御技術については同様に確立されていない。

【0011】しかしながら、階層化された複数のオブジェクトが管理される情報処理システム環境において、記憶されたオブジェクトデータの有効利用及び利用効率アップを図るためにには、前記した第1のロック要求の有ったオブジェクトとそのオブジェクトと親子関係にあるオブジェクトのみを、該第1のロック要求を行ったプロセスに対してのみに排他的に使用可能（アクセス可能）とするロック制御の実現が望まれていた。

【0012】本発明は、階層化された複数のオブジェクトの制御が可能な情報処理システム環境で、1つのオブジェクトに対する第1のロック要求がなされた場合に、そのオブジェクトと、そのオブジェクトと親子関係にある全てのオブジェクトのみに対して、アクセスの排他制御を行うロック制御を実現することを課題とする。

【0013】

【課題を解決するための手段】本発明では、階層化された複数のオブジェクトの記憶と制御及びそれらのオブジェクトを用いた処理が可能な情報処理システム環境において、1つのオブジェクトに対してロック要求がなされた場合に、前記ロック要求のあった前記オブジェクトと、前記オブジェクトが属する階層より下位の全ての階層の前記オブジェクトの全ての親オブジェクトと、前記オブジェクトが属する階層より上位の全ての階層の前記オブジェクトの全ての親オブジェクトのみに対して、ロックを行うことを特徴とする。

【0014】上記ロックの制御は、例えば、以下に述べる方法により実現される。まず、まだロックされていない1つのオブジェクト（以後、第1のオブジェクトと記述する）に対してロック要求がなされた場合に、階層構造全体の中でのその第1のオブジェクトが属する階層位置と、その第1のオブジェクト及びその第1のオブジェクトの全ての親オブジェクトの名称とを含む名称のロッ

クファイルを生成する。

【0015】そして、その後、その第1のオブジェクトの子オブジェクトの一つにロック要求が有った場合には、その子オブジェクトの親オブジェクトを階層の低い順から順次遡って行くことにより、前記ロックファイルの名称内の第1のオブジェクトの名称と一致する名称の親オブジェクトが見つかるので、その子オブジェクトはロックされていると判定される。

【0016】又、その第1のオブジェクト自体にロック要求が有った場合には、前記ロックファイルの名称内に、その第1のオブジェクトの名称が存在するので、その第1のオブジェクトはロックされていると判定される。

【0017】又、更に、その第1のオブジェクトの親オブジェクトの一つにロック要求が有った場合には、その親オブジェクトの名称が、前記ロックファイルの名称内に存在するので、その親オブジェクトはロックされていると判定される。

【0018】更に、クライアント/サーバ環境では、階層化された複数のオブジェクトの管理をサーバで一括して行い、クライアントからの該複数のオブジェクトへのロック要求に対して、そのロック要求に対するロックファイルの生成によるロック制御とロック判定もサーバで一括して行うようにする。

【0019】又、クライアント/サーバ環境で、あるクライアントにおいて複数のオブジェクトに対するロック要求や複数のオブジェクトについてのロック情報取得要求が発生した場合には、そのクライアントからサーバへの上記要求はパケット等により一括して行われる。

【0020】

【発明の実施の形態】以下に本発明の実施の形態につき図を用いて詳細に説明を行う。図1は、情報処理システム環境の一形態であるクライアント/サーバ環境における階層化された複数のオブジェクトのロック制御方法を説明する図である。

【0021】尚、階層化されたオブジェクトでは、親オブジェクトが異なる場合には同名のオブジェクト名が一つの階層に複数存在し得る。従って、ある階層のオブジェクトは、同名のオブジェクトが存在しない筆頭（最上位）の親オブジェクトまでのオブジェクトの親子関係の名称を情報として持つ。例えば、最上位の階層の親オブジェクト名が「ABC」で2番目の階層の「ABC」の子オブジェクト名が「PQR」で、3番目の階層の「PQR」の子オブジェクト名が「XYZ」とすると、オブジェクト「XYZ」には、「ABC/PQR/XYZ」という親子関係を示すオブジェクトの名称情報が含まれる。

【0022】情報処理システムにおいて、あるオブジェクトへのロック要求とアクセスの排他処理が終了する前に、そのオブジェクトやそのオブジェクトと親子関係に

あるオブジェクトに対して次のロック要求が来る場合としては、様々な形態が考えられるが、説明の便宜のために、図1のような複数のクライアント1～4等からサーバ5内の階層化された複数のオブジェクトA、B、C内の任意のオブジェクトにロック要求が有る場合を考える。

【0023】ネットワーク6には、クライアント1～4等とサーバ5が接続されてローカルエリアネットワーク(LAN)等を形成している。クライアント1～4等は、少なくとも図示されないキーボード等の入力装置とディスプレーやプリンタ等の出力装置を備え、ユーザーの要求内容を入力装置からネットワーク6を介してサーバ5へ伝送することができるようになっている。

【0024】サーバ5は、階層化された複数のオブジェクト(階層化複数オブジェクト)A、B、C等を記憶でき、それら階層化複数オブジェクト内のいずれかのオブジェクトに変更を含むアクセスによるロック要求があつた場合には、そのオブジェクトがロックされていない場合、そのオブジェクト及びそのオブジェクトと親子関係にあるオブジェクトがロック中であることを示すロックファイルを生成して記憶できるようになっている。

【0025】階層化複数オブジェクトAの中の(1)の位置にあるオブジェクトA-(1)に対する内容変更を含むアクセスによるロック要求7-1があつた場合を考えると、ロック要求7-1で排他される対象としては、オブジェクトA-(1)自身を含んでオブジェクトA-(1)の下層に含まれるオブジェクトA-(1)の子オブジェクト(1)、(n)、及び、オブジェクトA-(1)の上層の親オブジェクトA-(2)が挙げられ、それらについても他クライアント2～4等からの変更を含むアクセスによるロック要求の排他を行う必要がある。

【0026】そのようなアクセスによるロック要求の排他を実現するために、以下に述べる名称のロックファイルの生成及びロック判定を用いる。尚、以後、名称Dのオブジェクトを「オブジェクトD」と表記する。

【0027】今、オブジェクトA-(1)に対するロック要求が発生して、前記したようにオブジェクトA-(1)の親オブジェクトや子オブジェクトにロック要求が生じたとすると、以下のロックファイル名称(A)で示されたロックファイルを生成する。

【0028】但し、ここで、図1とは異なりオブジェクトA-(1)が含まれる階層化複数オブジェクトAの階層構造の階層の深さを「m」、ロック要求の生じたオブジェクトA-(1)の最上位階層の親オブジェクトから該オブジェクトA-(1)までの相対的な階層の深さを「n」とする。

【0029】

(A) = "P(n).P(m-1).…,P(2).P(1).A-(1)... …."

(A) : 生成するロックファイル名

A-(1) : ロック要求のあったオブジェクトの名称

P(i) : オブジェクトA-(1)を子オブジェクトとして含み、オブジェクトA-(1)に対してi階層分だけ上位の階層の親オブジェクトの名称

上記の生成するロックファイル名(A)において、ファイル名に含まれるビリオド「.」の数は、階層構造の階層の深さと同数の「m」個であり、名称A-(1)の後のビリオド「.」の数は、階層構造の全体の階層数「m」からオブジェクトA-(1)の最上位オブジェクトP(n)からの相対的な深さの階層数「n」を引いた「m-n」個である。

【0030】また、オブジェクトA-(1)にロックが掛かっているか否かの判定は、以下の名称(A)-(1)～(A)-(n+1)のロックファイルが存在すればロジックが掛かっており、存在しなければロックが掛かっていないと判定する。

【0031】

(A)-(1) : "P(n).…,P(2).P(1).A-(1).*.*. …."

(A)-(2) : "P(n).…,P(2).P(1).…,.."

(A)-(3) : "P(n).…,P(2).…,.."

:

(A)-(n+1) : "P(n).…,.."

ここで、(A)-(1)における"*"は、空文字を含む任意の文字列を意味する表現であり、従つて、(A)-(1)は、オブジェクトA-(1)の全ての子オブジェクトの名称が"*"に入る場合も含めて包括して表現したものであり、結果的に、オブジェクトA-(1)、及び、オブジェクトA-(1)の全ての子オブジェクトがロックされていることを示している。

【0032】(A)-(2)～(A)-(n+1)のロックファイル名は、オブジェクトA-(1)の各親オブジェクトのロックファイルの名称である。上記(A)-(1)～(A)-(n+1)のロックファイル名において、ファイル名に含まれるビリオド「.」の数は全て「m」である。(A)-(1)は、オブジェクトA-(1)またはその子オブジェクト全てのロックファイルの名称であり、(A)-(2)～(A)-(n+1)は、オブジェクトA-(1)の各親オブジェクトのロックファイルの名称である。

【0033】尚、上記ロックファイルの名称(A)を、オブジェクトA-(1)のロックファイルのレコードとして記録するようにしてもよい。この場合、該ロックファイルの名称は、所定の方法により定めればよい。

【0034】図1において、クライアント1からネットワーク6を介して、サーバ5内のオブジェクトA-(1)にロック要求7-1が行われ、オブジェクトA-(1)がロックされていなかった場合には、オブジェクトA-(1)と、その子オブジェクトA-(1)及びA-(n)と、オブジェクトA-(1)の親オブジェクトA-(2)をロックするために上記ロックファイル名(A)のロックファイルが生成されて記憶される。(この場合、「A-(2)、A-(1)…」の名称のロックファイルが生成される)

ここで、ロック要求7-1 の処理中、即ち、上記ロックファイル名(A)のロックファイルによりロックされている間に、クライアント2からサーバ5内のオブジェクトA-(n)にロック要求7-2 があった場合、オブジェクトA-(n)の名称を含むか、オブジェクトA-(n)の親オブジェクトの名称を含むロックファイルが存在する否かによりロック中であるか否かが判定される(存在していれば、ロック中であると判断される)。

【0035】このロックファイル名(A)のロックファイルによりオブジェクトA-(2)、A-(1)、A-(0)がロックされている状況で、クライアント2から、オブジェクトA-(n)にロック要求7-2 が来た場合、ロックファイル名(A)のロックファイルのみが存在していれば、オブジェクトA-(n)はロック中でないと判定される。そして、「A-(2)、A-(1)、A-(n)」の名称のロックファイルが生成される。

【0036】次に、クライアント2からオブジェクトB-(1)にロック要求7-3 が行われた場合には、オブジェクトB-(1)がロックされていなければ、オブジェクトB-(1)、オブジェクトB-(1)の親オブジェクトB-(3)、B-(2)がロック中であることを示す「B-(3)、B-(2)、B-(1)」というファイル名称(B)のロックファイルが生成されて記憶される。ここで、ロック要求7-3 による変更アクセス処理中、即ち、ロックファイル名(B)のロックファイルによりロックされている間に、他のクライアントから階層化複数オブジェクトB内のいずれかのオブジェクトに変更を含むアクセスがあった場合、ロックファイル名(B)のロックファイルの存在によりロック中であるか否かが判定される。

【0037】クライアント3からオブジェクトC-(1)にロック要求7-4 を行う場合には、オブジェクトC-(1)がロックされていなければ「C-(1)」というロックファイル名(C)のロックファイルが生成されて記憶される。

【0038】そして、ロック要求7-4 による変更アクセス処理中、即ち、ロックファイル名(C)のロックファイルによりロックされている間に、他のクライアントから階層化複数オブジェクトC内のいずれかのオブジェクトに変更を含むアクセスがあった場合、ロックファイル名(C)のロックファイルの存在によりロック中であるか否かが判定される。

【0039】この図1の階層化複数オブジェクトA、B、C内の各オブジェクトと、そのロックファイル名称を具体的にして、更に詳しく示した図が図2である。図2では、各階層化複数オブジェクトA、B、Cの最上位の階層の親オブジェクト名称は、それぞれEDBID1、EDBID2、EDBID3(EDBID : Engineering Data-BaseID)の名称で示され、その下の層のオブジェクトの名称はそれぞれ文書1、文書2の名称で示され、最下層のオブジェクトの名称はシート1、シート2、シート3の名称で示され

ている。

【0040】階層構造の深さ「m」は、「3」である。従って、単純にシート1、シート2あるいは文書1、文書2といったロックファイルの名称設定をすると、一つの階層化複数オブジェクトの中に複数のオブジェクトシート1、シート2等が存在し、又、各階層化複数オブジェクトA、B、C内に、それぞれ、オブジェクト文書1、文書2が存在しているため、どのオブジェクトを指すのか特定できないので、ロックファイルの名称には最上位の階層の親オブジェクトの名称まで遡った名称が記述される。

【0041】尚、以下の説明においては、例えば、オブジェクトA-(2)に含まれる一階層下の子オブジェクトA-(1)を表す場合には、階層関係を含めて「A-(2) / A-(1)」の様に表すこととする。又、クライアント1～4からサーバ5にロック要求又はロック情報取得要求が送られる場合、ロック対象又はロック情報取得対象のオブジェクトは、この表記方法によりサーバ5に通知される。

【0042】まず、階層化複数オブジェクトAの場合には、2番目の層のオブジェクト文書2が、ロック要求7-1 の対象であるので、「EDBID1/文書2」のオブジェクトにロック要求が発生することになる。

【0043】その結果、「EDBID1/文書2」のオブジェクトがロック中かを後述する調査方法で調べる。もし、調査結果がロック中でないならば、前述した方法によりロックファイル名称「EDBID1. 文書2..」の名前のロックファイルを生成して「EDBID1」、「EDBID1/文書2」、「EDBID1 / 文書2 / シート1」と「EDBID1/文書2 / シート2」のオブジェクトをロック状態にする。

【0044】このように「EDBID1. 文書2..」ロックファイル作成することによって、点線aによって囲まれた4個のオブジェクト(「EDBID1」、「EDBID1/文書2」、「EDBID1/文書2 / シート1」と「EDBID1/文書2 / シート2」)がロック中になり、後述するロック状態判定の方法において、これらのオブジェクトがロック中であることが判定できるようになる。

【0045】次に、階層化複数オブジェクトBの場合には、最下層のオブジェクトシート2が、ロック要求7-3 の対象であるので、「EDBID2/文書1 / シート2」のオブジェクトにロック要求が発生する。

【0046】その結果、「EDBID2 / 文書1 / シート2」のオブジェクトがロック中かを後述する調査方法で調べる。もし、調査結果がロック中でないならば、前述した方法によりロックファイル名称「EDBID2. 文書1. シート2..」の名前のロックファイルを生成して「EDBID2」、「EDBID2 / 文書1」と「EDBID2 / 文書1 / シート2」のオブジェクトをロック状態にする。

【0047】このようにロックファイルを作成することによって、点線bによって囲まれた3個のオブジェクト

(「EDBID2」、「EDBID2/文書1」と「EDBID2/文書1/シート2」)がロック中になり、後述するロック状態判定の方法において、これらのオブジェクトがロック中であることが判定できるようになる。

【0048】更に、階層化複数オブジェクトCの場合は、最上位の階層のオブジェクトEDBID3が、ロック要求7-4の対象であるので、「EDBID3」のオブジェクトにロック要求が発生する。

【0049】その結果、「EDBID3」のオブジェクトがロック中かを後述する調査方法で調べる。もし、調査結果がロック中でないならば、前述した方法によりロックファイル名称「EDBID3...」の名前のロックファイルを生成して「EDBID3」、「EDBID3/文書1」、「EDBID3/文書2」、「EDBID3/文書1/シート1」、「EDBID3/文書1/シート2」、「EDBID3/文書2/シート1」のオブジェクトをロック状態にする。

【0050】このようにロックファイルを作成することによって、点線cによって囲まれた階層化複数オブジェクトCの全てのオブジェクトである6個のオブジェクト(「EDBID3」、「EDBID3/文書1」、「EDBID3/文書2」、「EDBID3/文書1/シート1」、「EDBID3/文書1/シート2」と「EDBID3/文書2/シート1」)がロック中になり、後述するロック状態判定の方法において、これらのオブジェクトがロック中であることが判定できるようになる。

【0051】次に、オブジェクトがロック中であるか否かの調べ方を説明する。階層化複数オブジェクトAの場合は、点線aで囲まれたオブジェクトがクライアント1からのロック要求7-1でロックされている場合である。ここで、他のクライアント2等から階層化複数オブジェクトAの「EDBID1/文書2」のオブジェクトに対して、ロック要求があったか、ロックされているかの調査要求(ロック情報取得要求)があったとする。

【0052】①まず、「EDBID1.文書2.*.」の名称の「EDBID1/文書2」自体またはその子オブジェクトのロックファイルが存在するか否かを調べる。
②次に、「EDBID1...」の名称の「EDBID1/文書2」の親オブジェクトのロックファイルが存在するか否かを調べる。

【0053】③上記①、②の調査で全くログファイルが存在しなければロック中ではないと判定される。また、上記①、②の調査でロックファイルが1つでも存在した場合はロック中であると判定される。

【0054】階層化複数オブジェクトBの場合は、点線bで囲まれたオブジェクトがクライアント2からのロック要求7-3でロックされている場合で、他のクライアント1や3等から階層化複数オブジェクトBの「EDBID1/文書2/シート2」に対して、ロック要求があったか、ロック情報取得要求があったとする。

【0055】この場合は、「EDBID1/文書2/シート

2」についてのロック判定を行う。

①まず、「EDBID1.文書2.シート2.」の名称の「EDBID2/文書1/シート2」自体のロックファイルが存在するか否かを調べる。

【0056】②次に、「EDBID2.文書1...」または「EDBID2...」の名称の「EDBID2/文書1/シート2」の親オブジェクトのロックファイルが存在するか否かを調べる。

③上記①、②の調査で全くロックファイルが存在しなければロック中ではないと判定される。また、上記①、②の調査でロックファイルが1つでも存在した場合はロック中であると判定される。

【0057】階層化複数オブジェクトCの場合は、点線cで囲まれたオブジェクトがクライアント3からのロック要求7-4でロックされている場合で、他のクライアント1や2等から階層化複数オブジェクトCの「EDBID3」に対して、ロック要求があったか、ロックされているかのロック情報取得要求があったとする。

【0058】この場合は、「EDBID3」についてのロック判定を行う。

①「EDBID3.*.*.」の名称のロックファイルが存在するか否かを調べる。

②上記の調査で全くロックファイルが存在しなければロック中ではないと判定される。また、上記調査でロックファイルが1つでも存在した場合はロック中であると判定される。

【0059】但し、上記において、「*」は空文字を含む任意の文字列を表す。図3は、サーバ5における、ロック制御の動作のフローチャートである。ステップS1では、クライアントのいずれかからネットワーク6経由で要求された今回のロック要求の対象のオブジェクト(以下、対象オブジェクトと表現する)の名称を、サーバ5で受け付ける。

【0060】ステップS2では、受け付けた対象オブジェクトがサーバ5の中に存在するか否かの調査を行う。ステップS3では、ステップS2の調査結果を基に対象オブジェクトがサーバ5中に存在したか否かの判断を行い、対象オブジェクトが存在する場合にはステップS4に進み、対象オブジェクトが存在しない場合にはステップS7へ進む。

【0061】ステップS4では、受け付けた対象オブジェクトの名称と現在存在している全てのロックファイルの名称から、上記図2を用いて説明した当該ロックファイルが存在するか否かの方法によりロック判定を行い、対象オブジェクトがロック中であるか否かの調査を行う。調査した結果、ロック中であれば、ステップS7へ進み、ロック中でなければステップS5へ進む。

【0062】ステップS5では、上記のステップS4の調査結果で、今回ロック要求でロック制御処理される対象のオブジェクトの中に、既にロックされているオブジ

エクトが無いことがわかったので、今回のロック要求により、ロック制御を行うためのロックファイルを生成する。

【0063】ステップS6では、今回のロック要求に対する、上記図2で説明した対象オブジェクトのロック制御処理を行う。ステップS7では、今回のロック要求を依頼してきたクライアントに対し、対象オブジェクトに対するロックが成功したか否かを、ネットワーク6を介して通知する。

【0064】図4は、図3のステップS1で受け付けたロック要求の対象オブジェクトが複数になり、3層の階層化複数オブジェクトにおいて第1層と第2層の親オブジェクト名が共通の第3層目の複数のオブジェクトを一括ロック要求する場合のサーバ5のロック制御を、クライアント側から見た場合のフローチャートである。

【0065】複数のオブジェクトに対する一括のロック要求が発生した場合には、全てのオブジェクトがロックされたことを保証するために、1つでもオブジェクトがロックできなかった場合は全てのロックを取り消すallモードと、ロック可能なオブジェクトのみロックするpossibleモードを用意する。

【0066】ステップS11では、いずれかのクライアントからサーバ5の4個のオブジェクト(「EDBID5/文書1/シート1」、「EDBID5/文書1/シート2」、「EDBID5/文書1/シート3」、「EDBID5/文書1/シート5」)に対して一括ロック要求が出され、サーバ5ではこの一括ロック要求に対しで図3のステップS1～ステップS5の処理を行う。

【0067】ステップS12は、上記ステップS1～S5の処理で全ての対象オブジェクトがロック中でなかつたため、全ての対象オブジェクトに対してロックファイルの生成が行われ、全ての対象オブジェクトのロックに成功した場合である。

【0068】ステップS14は、ステップS12の全ての対象オブジェクトのロック成功の結果を受けて一括ロック要求に対する処理が完了したことを示す。ステップS13は、ロック中のオブジェクト(「EDBID5/文書1/シート3」)が1個だけ有ったため、そのオブジェクトのロックファイルの生成が行われず、そのオブジェクトのみについてはロックに失敗した場合である。この場合は、全てのオブジェクトに対しロックを行わずにオブジェクトの変更処理を中止するか、ロック可能なオブジェクトのみにロックを行って該オブジェクトのみ変更処理を行うかの選択をユーザーの設定したロックモードに従って行う。ステップS13の場合はクライアントからの一括ロック要求時に一緒に出されたロックモードの指示に従って、ステップS15またはS16へ進む。

【0069】ステップS15は、ユーザーがpossibleモードを選択した場合の処理であり、ロック失敗したオブジェクト(「EDBID5/文書1/シート3」)を除いて今

回ロックされたオブジェクト(「EDBID5/文書1/シート1」、「EDBID5/文書1/シート2」、「EDBID5/文書1/シート5」)のみロック完了したことを示す。

【0070】ステップS16は、ユーザーがallモードを選択した場合の処理であり、ロックに成功した対象オブジェクト(「EDBID5/文書1/シート1」、「EDBID5/文書1/シート2」、「EDBID5/文書1/シート5」)のロック処理を解除、即ち、一度生成した該対象オブジェクトのロックファイルを削除して、今回の一括ロック処理を取り消したこと示す。

【0071】上記のように図4から、全ての対象オブジェクトがロックされたことを保証するために、一つでも対象オブジェクトがロックできなかった場合は全ての対象オブジェクトに対してロックを行わないallモードと、ロック可能な対象オブジェクトのみロックするpossibleモードを、クライアントに対して用意することで、複数のオブジェクトに対する一括のロック要求が発生した場合にも対処できる。

【0072】図5は、クライアント/サーバ環境において、一つの統括サーバ5がロック管理(上記ロックファイルの生成・削除、ロック情報取得)を一元的に統括するシステムの構成図である。クライアント1からのロック情報取得請求10、及び、クライアント3からのロック要求11は統括サーバ5に送られ、これらの要求10、11は該統括サーバ5により本実施形態のロック制御方法で統括して処理され、統括サーバ5上にロックファイルが生成される(ロック要求11の場合)。従って、親子関係のオブジェクト同士の関係に矛盾を起さずに、クライアント/サーバ環境上で階層化複数オブジェクトのロック管理を一元的に行うことができる。

【0073】例えば、クライアント1において、オブジェクトAをエディタ等で更新したい場合を考える。その場合には、まず、オブジェクトAに関するロック情報を取得するよう統括サーバ5にロック情報取得要求10を送る。

【0074】統括サーバ5では、このロック情報取得要求10を受け取ると、オブジェクトAがロックされているか否かを判定する。判定方法としては、オブジェクトA、及び、その親子関係にあるオブジェクトのロックファイルの存在を調べる。そして、その判定結果をクライアント1に返す。

【0075】クライアント1では、返ってきた判定結果で、オブジェクトAがロックされているということであれば、読み込みモードでのみオブジェクトAにアクセス可能ということになります。逆に判定結果で、オブジェクトAがロックされていないということになれば、読み取りモードの他に更新モードでもオブジェクトAとアクセスが可能ということになります。

【0076】クライアント1から更新モードでオブジェクトAとアクセスする場合、更新中に他のアクセスに

よりオブジェクトAが更新されないようにするために、オブジェクトAをロックする必要があります。そのためにはロック要求を統括サーバ5に送る必要があります。

【0077】統括サーバ5では、ロック要求を受けて、オブジェクトAのロックファイルを統括サーバ5上に生成します。オブジェクトAのロックファイルの生成後は、クライアント1でオブジェクトAを自由に更新が可能になります。

【0078】クライアント1によるオブジェクトAの更新処理の終了後、ロックを続ける必要が無くなると、クライアント1は統括サーバ5にオブジェクトAのロック解除要求を送信する。統括サーバ5では、このロック解除要求を受けて、オブジェクトAのロックファイルを削除する。

【0079】上記で示した手順は、最初にクライアント1から統括サーバ5に対してオブジェクトAのロック情報取得要求10を送り、統括サーバ5での判定結果をクライアント1で受けて、オブジェクトAがロックされていないという場合にロック要求を送るというステップになっていますが、最初のロック情報取得要求10を送り判定結果を受け取るステップを省略して、図5におけるクライアント3からのロック要求11のように、最初にロック要求11を送ることも可能である。

【0080】その最初にロック要求を送信可能な理由としては、統括サーバ5では、ロック要求11を受けると、その度毎にロック要求のあったオブジェクトがロック中であるか否かの判定を行い、その判定結果によりロック要求の有ったオブジェクトのロックを行うか否かを決定し、そのオブジェクトのロックの成否の情報はクライアント側に返されるようになっているためである。

【0081】図6は、クライアント1～4等側で、上記した複数のオブジェクトについて一括のロック要求が発生した場合に、ネットワーク上のトラフィック負荷の低減のため、統括サーバ5に一括ロック要求をパケット情報9で送る場合を示す。

【0082】この図6の場合には、1個のパケット情報9の中には、そのパケット情報がサーバ5へのロック要求の情報であることを示すロック要求コマンド9-0と、上記possibleモード又はallモードのユーザーが選択したロックモードが設定されるロックモード情報9-1と、デリミタ9-2、9-4、9-6、9-8、9-9とこれらのデリミタで区切られた間に入る各階層毎のオブジェクトの名称情報9-3、9-5、9-7が入る。

【0083】3層の階層化複数オブジェクトの第3層目のロック要求がなされた複数の対象オブジェクト(「EDBID/文書ID/シート1」、「EDBID/文書ID/シート2」、…、「EDBID/文書ID/シートn」)は、パケット情報9の中では第1層目と第2層目は共通化されて「ロックモード: \$n : EDBID : \$n : 文書ID : \$n : シー

ト1、シート2、…、シートn : \$n : \$n : 」というように送られる。

【0084】例えば、図4の一括ロック要求の最初のオブジェクト「EDBID5/文書1/シート1」(階層構造が3層でその3層目のオブジェクト)が1個だけパケットに入るとすると、「ロックモード: \$n : EDBID5 : \$n : 文書1 : \$n : シート1」となる。そしてその他のオブジェクトのシート2、3、5についても、個々にパケットを作成して、それらのパケットを順次統括サーバ5に送出しなければならない。これが、図4のステップS11の一括ロック要求のあった4個のオブジェクトを一括して送る場合には、「EDBID5/文書1」までは共通であるので、「ロックモード: \$n : EDBID5 : \$n : 文書1 : \$n : シート1、シート2、シート3、シート5」となり、1個のパケットのみ統括サーバ5に送信すればよいので、ネットワーク6上のトラフィック負荷を低減することができる。

【0085】送られたパケット情報9は、統括サーバ5で個別のオブジェクトを示す形式「EDBID/文書ID/シート1」「EDBID/文書ID/シート2」…「EDBID/文書ID/シートn」に変換されて各オブジェクトのロック制御処理が行われ、ロックファイルを生成させる場合には、上記オブジェクトを示す形式が更にそのオブジェクトのロックファイルの名称を示す形式に変換されて、該名称のロックファイルが生成される。

【0086】このように、各クライアントがサーバ5に対して、パケット情報9により一括してロック要求のあった複数のオブジェクトの情報を送ることで、ネットワーク6上のトラフィック負荷を低減することができる。

【0087】図7は、情報処理システム環境における、本実施形態の階層化された複数のオブジェクトのロック制御方法を実現するプログラムを記憶媒体に記憶させた実施の形態の構成図である。

【0088】情報処理装置21は、図1や図5のサーバ5に相当し、メインフレーム、ワークステーション、あるいは、パソコン等の情報処理が可能な装置であり、本実施形態のロック制御処理を実行できるプログラム(データ)26を記憶できる記憶装置24と、記憶内容や処理結果等を出力可能な出力装置としてのディスプレイ23、可搬記憶媒体31を駆動するための媒体駆動装置25、図示しない入力装置や処理装置等を備える。

【0089】プログラム(データ)提供者41は、情報処理装置21とは、回線51により接続されており、プログラム(データ)提供者41内に記憶されたプログラム(データ)42を回線51を介して情報処理装置21内の記憶装置24に転送してプログラム(データ)26として記憶させる事ができる。

【0090】可搬性記憶媒体31は、例えば、プログラム(データ)34を記憶したCD-ROM32や、プロ

グラム（データ）35を記憶したフロッピーディスク3のように、情報処理装置21で処理されるプログラムやデータを記憶しており、情報処理装置21の媒体駆動装置25に挿入されて、媒体駆動装置25によりプログラム（データ）34、35を読み出されことで、情報処理装置21内の記憶装置24にプログラム（データ）26として記憶させる事ができる。

【0091】従って、当初は、プログラム（データ）提供者41から送信されるプログラム（データ）42である場合でも、過般記憶媒体31にCD-ROM32のプログラム（データ）34やフロッピーディスク33のプログラム（データ）35として記憶されている場合でも、最終的には、情報処理装置21の記憶装置24に記憶されたプログラム（データ）26として実施されるので、プログラム（データ）としてはその記憶される媒体による限定を受けず同様な取り扱いが可能となる。

【0092】

【発明の効果】本発明によれば、階層化された複数のオブジェクトにおいて、ロック要求の有ったオブジェクトと該オブジェクトと親子関係にあるオブジェクトのみをロックするので、その後のその階層化複数オブジェクト内の該オブジェクトと親子関係には無いオブジェクトに対するアクセス要求は可能となるので、オブジェクトへのアクセス効率を下げることの無い有効なアクセスの排他制御（ロック管理）が可能になる。

【0093】更に、必要な親オブジェクトや子オブジェクトのみへのロックが行われるので、ロック要求による処理中のオブジェクトと親子関係のあるオブジェクトへの新たなロック要求が発生した場合に両オブジェクト間の関係に矛盾を生じさせることなく、合理的かつ簡潔なロック制御（排他制御）を行うことが可能になる。

【0094】また、一括ロック要求のあったオブジェクトの中に親、又は、子オブジェクトのいずれかのオブジェクトが既にロックされているオブジェクトが有る場合にも、ロック要求の有った全てのオブジェクトのロックを取り消すか、又は、まだロックされていないオブジェ

クトのみをロックするかを選択できるので柔軟なロック制御が可能となる。

【0095】更に、クライアント／サーバ環境で、サーバが、階層化された複数のオブジェクトのロック制御を一元的に統括して行うことで、クライアント／サーバ環境におけるロック制御が簡潔になり、各オブジェクト間の変更による矛盾が無くなり、更に、クライアントから複数のオブジェクトのロック要求を一括してパケットにより受け付けることで、ネットワークにおける伝送負荷の低減が可能になる。

【図面の簡単な説明】

【図1】本発明の一実施形態のクライアント／サーバシステムの全体構成を示す図である。

【図2】上記実施形態による階層化された複数のオブジェクトのロック制御方法を説明する図である。

【図3】サーバのロック制御処理を説明するフローチャートである。

【図4】複数のオブジェクトの一括ロック要求が行われた場合の、allモードとpossibleモードにおけるロック処理を説明するフローチャートである。

【図5】クライアント／サーバシステムにおけるサーバによるロック情報の一元管理方法を示す図である。

【図6】サーバに一括ロック要求するパケット情報の形式の一例を示す図である。

【図7】記憶媒体に記憶させた本発明の実施形態のプログラムを実施する場合の構成を説明する図である。

【符号の説明】

1～4 クライアント

5 サーバ

30 6 ネットワーク

7～1～7～4 ロック要求

9 パケット情報

10 ロック情報取得要求

11 ロック要求

A、B、C 階層化複数オブジェクト

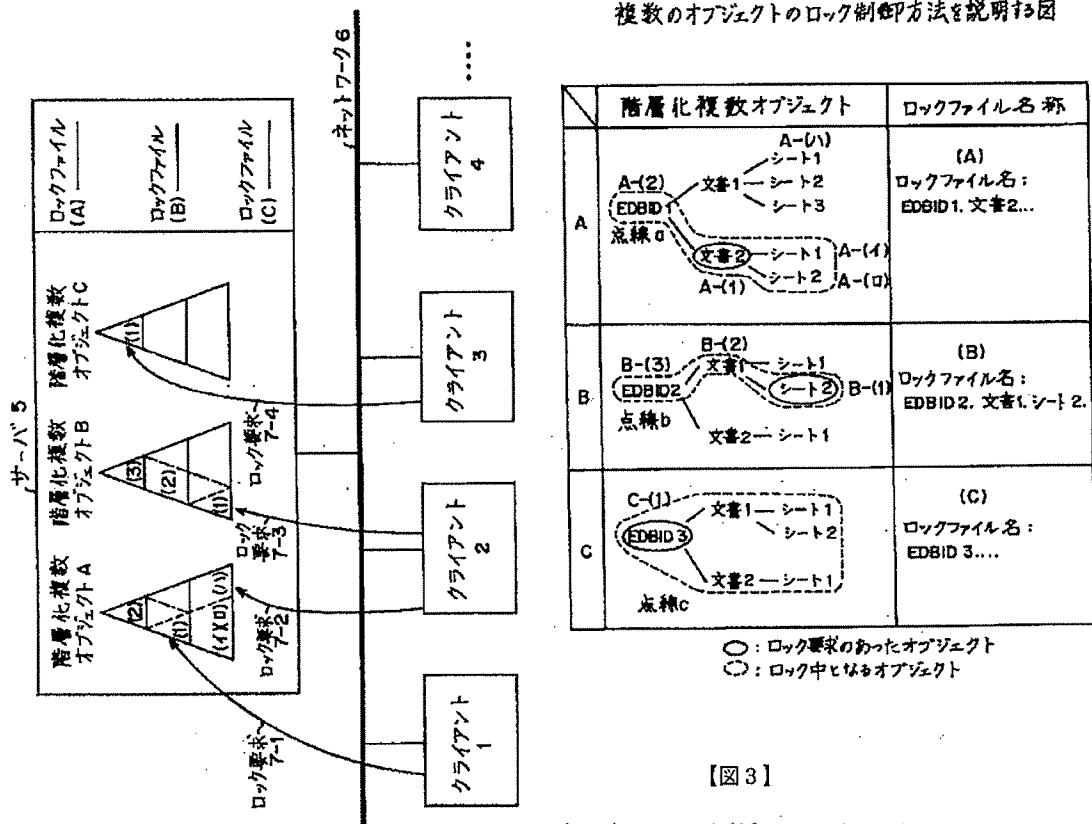
(A)、(B)、(C) ロックファイル名

[1]

[図2]

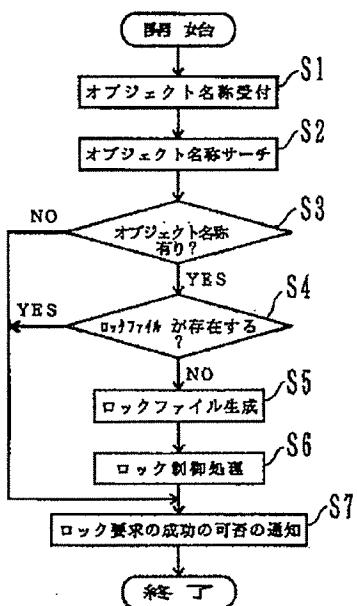
全体構成を示す図

本発明の実施形態による階層化された複数のオブジェクトのロック制御方法を説明する図

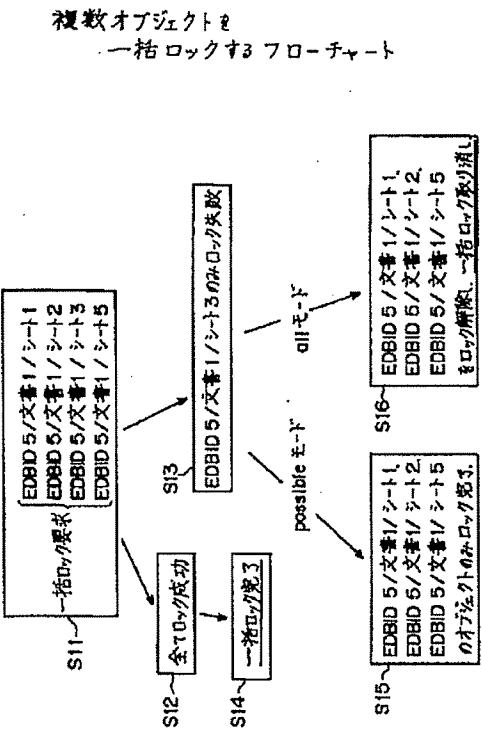


[図3]

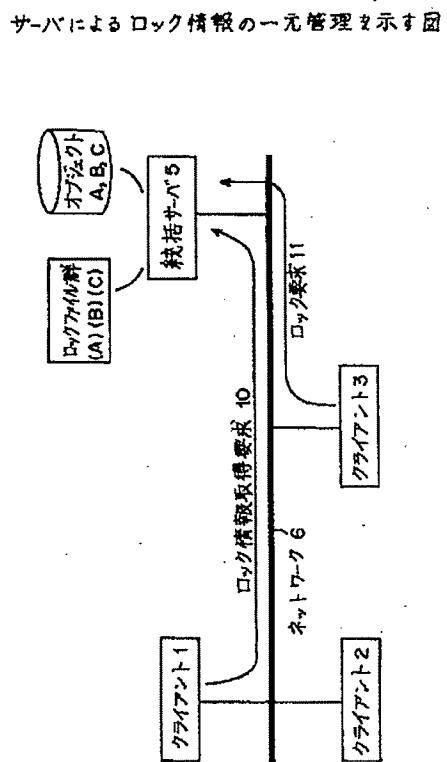
サーバでのロック制御フロー・チャート



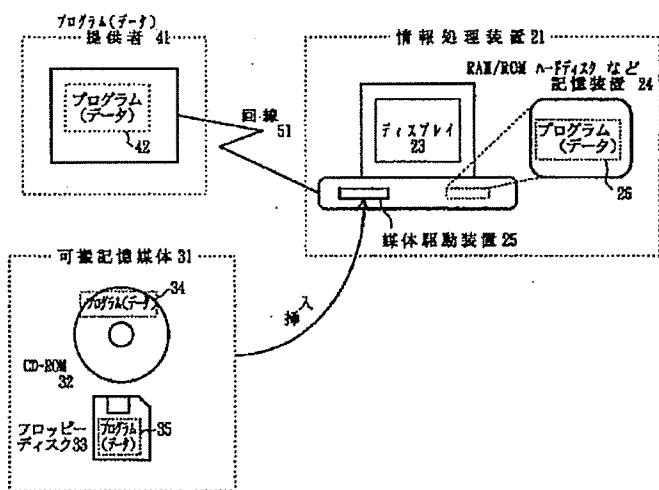
【図4】



【図5】

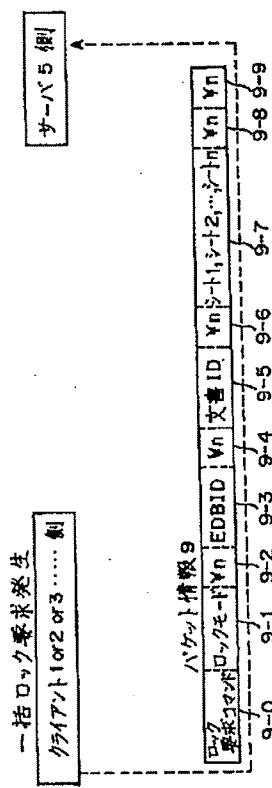


【図7】



【図6】

サーバに一括ロックするパケット情報を示す図



(11) Japanese Patent Laid-Open No. 11-85597

(43) Laid-Open Date: March 30, 1999

(21) Application Number: 9-239742

(22) Filing Date: September 4, 1997

(71) Applicant: FUJI ELECTRIC CO LTD

FF C: KK

(72) Inventor: YUKIO TAKANO

(72) Inventor: SHINICHIRO KOYAMA

(54) [Title of the Invention]

LOCK CONTROLLING METHOD ON LAYERED OBJECTS

[Abstract]

[Problem to be Solved]

To provide a method for implementing lock controlling to lock only an object to which a lock request is issued and all objects that have parent-child relationship with the object when the lock request is issued to the object in lock controlling to layered objects.

[Solution]

In an information processing system environment in which layered objects A, B and C can be controlled, when a client 1 issues a lock request to an object A-(1) and the object A-(1) is not locked, a lock file (A) including the name of the object A-(1), its layer in a layer structure and the names of objects A-(2) that have parent-child relationship with the object A-(1) and are present at layers higher than

that of the object A-(1) is generated, and the object A-(1), all the objects A-(2) that have parent-child relationship with the object A-(1) and are present at layers higher than that of the object A-(1), and all the objects A-(I), A-(RO) that have parent-child relationship with the object A-(1) and are present at layers lower than that of the object A-(1) are locked.

[Claims for the Patent]

[Claim 1]

A lock controlling method on layered objects in an information processing system environment in which layered objects can be stored and controlled and control using the objects is possible, characterized in that

when a lock request is issued to an object, only said object, to which said lock request is issued, all child objects of said object which are in all layers lower than the layer of said object, and all parent objects of said object which are in all layers higher than the layer of said object are locked.

[Claim 2]

The lock controlling method on layered objects according to claim 1, characterized in that

for said object to which said lock request is issued, said lock controlling is performed by generation and deletion of a lock file,

wherein said lock file includes:

first information indicating at least a layer to which said object belongs among the layers of said layered objects; and

second information on a name of said object and names of all parent objects of said object.

[Claim 3]

The lock controlling method on layered objects according to claim 2, characterized in that

said lock file further includes third information indicating the number of layers of a layer structure of said layered objects.

[Claim 4]

The lock controlling method on layered objects according to claim 3, characterized in that

said first, second and third information are set as a name or a record of said lock file.

[Claim 5]

The lock controlling method on layered objects according to claim 4, characterized in that

the name or the record of said lock file includes a character string wherein the name of said object to which a lock request is issued and the names of all parent objects of the object are arranged in the descending order from an object at the highest layer separated by a particular symbol, and the name of the object to which a lock request is issued that is followed by said particular symbol is followed by the same number of said particular symbols as the number

of all layers lower than the layer to which the object to which a lock request is issued belongs;

said first information is indicated by the order of said particular symbol appeared in the character string, wherein said particular symbol follows said object to which a lock request is issued; and

said second information is indicated by each object name in the character string;

an object name arranged at the end of the character string expresses the name of said object to which a lock request is issued;

other object names arranged in the character string express the names of said all parent objects; and

said third information is indicated by the total number of said particular symbol in the character string.

[Claim 6]

The lock controlling method on layered objects according to claim 2, 3, 4 or 5, characterized in that

when a new lock request is issued, whether an object to which the new lock request is issued has been locked or not is determined by searching for said lock file indicating that the object is locked based on the name of the object to which a lock request is issued.

[Claim 7]

A lock controlling method on layered objects in an information processing system environment in which layered

objects can be stored and controlled and control using the objects is possible, characterized in that

when a collective lock request is issued to objects, locking is attempted to only all child objects of each of objects at all layers lower than the layer to which said object to which a lock request is issued belongs and all parent objects of each of said objects at all layers higher than the layer to which said object to which a lock request is issued belongs; and

if there is an object that cannot be locked, locking is not attempted to all of said objects to which a lock request is issued.

[Claim 8]

A lock controlling method on layered objects in an information processing system environment in which layered objects can be stored and controlled and control using the objects is possible, characterized in that

when a collective lock request is issued to objects, locking is attempted to only all child objects of each of objects at all layers lower than the layer to which said object to which a lock request is issued belongs and all parent objects of each of said objects at all layers higher than the layer to which said object to which a lock request is issued belongs; wherein

a locked object is excluded and only objects which can be locked is locked.

[Claim 9]

A lock controlling method on layered objects,
characterized in that

when a collective lock request is to be issued to objects,
either the lock controlling method according to claim 7 or
the lock controlling method according to claim 8 is specified
when said lock request is issued.

[Claim 10]

The lock controlling method on layered objects according
to any one of claim 1 to claim 9, characterized in that
lock controlling on layered objects in response to said
lock request is collectively performed by a server in the
client/server environment.

[Claim 11]

The lock controlling method on layered objects according
to claim 10, characterized in that

when a lock request to objects or a request to determine
whether objects can be locked or not occurs at the client
side, the request is collectively left to said server.

[Claim 12]

A lock controlling device on layered objects that can
lock an object when a lock request is issued to the object
in an information processing system environment in which
layered objects can be stored and controlled and control
using the objects is possible, characterized in that said
device comprises:

means for generating/deleting a lock file for controlling
said locking by generation and deletion of a lock file, wherein

said lock file includes: first information indicating at least a layer to which said object belongs among the layers of said layered objects; and second information on a name of said object and names of all parent objects of said object;

means for determining whether an object to which a new lock request is issued has been locked or not by searching for said lock file indicating that the object is locked based on the name of the object to which a lock request is issued when the new lock request is issued; and

means for controlling to cause said means for generating/deleting a lock file to generate a lock file corresponding to said object to which a lock request is issued when the object can be locked based on the determination from said means for determining.

[Claim 13]

A computer readable recording medium that stores a program for causing a computer to execute the processes of:

when a lock request is issued to an object in an information processing system environment in which layered objects can be stored and controlled and control using the objects is possible,

generating a lock file including first information indicating at least a layer to which said object belongs among the layers of said layered objects; and second information on a name of said object and names of all parent objects of said object; and

causing exclusion of other lock requests to be performed by the lock file.

[Detailed Description of the Invention]

[0001]

[Field of the Invention]

The present invention relates to a method for controlling lock (access exclusion) of layered objects in information processing system environment.

[0002]

[Conventional Art]

As one of conventional methods for controlling lock of layered objects in information processing system environment, lock (exclusion) controlling is known. With the lock (exclusion) controlling, when a first request to exclude an access including a change in descriptions of an object (a lock request: hereinafter referred to as a first lock request) is processed and a second request to exclude an access including a change in descriptions of the object (hereinafter referred to as a second lock request) is received, the second lock request is rejected until the first lock request finishes.

[0003]

Among ways of locking, lock controlling to lock the entire layered objects in accordance with the first lock request and exclude the second and after lock requests; and controlling to lock only an object, to which a request to exclude an access is issued, in accordance with the first

lock request and exclude the second and after lock requests are known.

[0004]

With the former exclude controlling to lock the entire layered objects, the second and after lock requests are excluded from causing objects with no parent-child relationship with the object, to which the first lock request is processed to be processed, until processing on the object in accordance with the first lock request finishes. Then, the second and after lock requests have no choice but to wait until the processing in accordance with the first lock request finishes.

[0005]

With the latter exclude controlling to lock only the object, to which a lock request is issued, even when the processing is performed in accordance with the first lock request, processing in accordance with the second or after lock requests is available for an object other than the object that is processed. If an object that has parent-child relationship with the object that is processed in accordance with the first lock request is processed in accordance with the second or after lock request while the object that is processed in accordance with the first lock request is processed, however, the parent-child relationship among the objects may be inconsistent.

[0006]

The term parent-child relationship in the specification of the present invention means relationship such as parent, grandparent, great grandparent, child, grandchild, great grandchild to an object. For example, a parent object means all the objects in the layers upper than that of a subject object that has relationship such as parent, grandparent, great grandparent to the subject object by inheritance.

[0007]

The term child object means all the objects in the layers lower than that of a subject object that has relationship such as child, grandchild, great grandchild to the subject object by inheritance. Thus, in the specification of the present invention, such terms as parent-child relationship, parent object, and child object mean the abovementioned broad senses of parent-child relationship, a parent object, a child object.

[0008]

[Problems to be Solved by the Invention]

Neither a method for indicating the place of an object in the layers of objects nor a method for indicating an object that has parent-child relationship with the subject object has been developed. Thus, without any method for determining an object that has parent-child relationship with the subject object, it has been impossible to determine an object that has parent-child relationship with the subject object and lock only the subject object and the object that has parent-child

relationship with the subject object when the subject object is changed.

[0009]

Any effective and simple controlling technique to lock an object to which a lock request is issued and an object that has parent-child relationship with the subject object has not been established. When a lock request is issued to an object among layered objects, there has been no choice but to lock the entire layered objects at the cost of efficiency in object data usage or to lock only the object to which a lock request is issued at the risk of occurrence of inconsistency in parent-child relationship among the layered objects.

[0010]

Any effective and simple controlling technique to lock only an object to which a lock request is issued and an object that has parent-child relationship with the subject object has not been established in client/server environment, either.

[0011]

To make the most of the stored object data and improve the efficiency in the stored object data usage in an information processing system environment in which layered objects are managed, such lock controlling to make only a process, for which the first lock request is issued, available (allowed to access) exclusively to the object to which the first lock request is issued and an object that has parent-child relationship with the subject object has been desired.

[0012]

The present invention intends to provide lock controlling for performing exclusive access controlling only on an object to which a first lock request is issued and all the objects that have parent-child relationship with the subject object in an information processing system environment in which layered objects can be controlled.

[0013]

[Means for Solving the Problems]

The present invention is characterized in that, in an information processing system environment in which layered objects can be stored and controlled and processing using the objects is possible, when a lock request is issued to an object, only the object, to which the lock request is issued, all child objects of the object which are in all layers lower than the layer of the object, and all parent objects of the object which are in all layers higher than the layer of the object are locked.

[0014]

The lock is controlled by the method to be described below. First, when a lock request is issued to an object that has not been locked (hereinafter referred to as a first object), a lock file under a name including a position of the first object layer in the layer structure and names of the first object and all parent objects of the first object is generated.

[0015]

When a lock request is issued to one of the child objects of the first object afterwards, parent objects of the child object are serially traced from the lowest layer and parent objects whose names match the first object name in the names included in the lock file are found. Accordingly, the child object is determined as locked.

[0016]

When a lock request is issued to the first object, it is determined that the first object is locked, as the first object name is in the names included in the lock file.

[0017]

When a lock request is issued to one of the parent objects of the first object, it is determined that the parent object is locked, as the parent object name is in the names included in the lock file.

[0018]

In client/server environment, a server collectively manages layered objects and collectively performs lock controlling and lock determination on the objects in response to a lock request issued from the client to the objects by generating a lock file for the lock request.

[0019]

When a lock request to objects or a request to obtain lock information for objects occurs at a client in client/server environment, the request from the client to the server is collectively issued by packet or the like.

[0020]

[Embodiments of the Invention]

Now, an embodiment in accordance with the present invention will be described in detail below with reference to the drawings. Figure 1 is a diagram illustrating a lock controlling method on layered objects in client/server environment which is an aspect of information processing system environment.

[0021]

In the layered objects, the same object name may be included in a layer if their parent objects are different. An object at a certain layer has information on names indicating parent-child relationship among objects through to a parent object at the top (the highest) layer that includes no object with the same name. Specifically, when the name of a parent object at the highest layer is 'ABC', the name of a child object of 'ABC' at the second highest layer is 'PQR', and the name of a child object of 'PQR' at the third highest layer is 'XYZ', the object 'XYZ' includes object name information of 'ABC/PQR/XYZ' indicating the parent-child relationship thereof.

[0022]

Various aspects can be assumed for the case where a lock request for a first object or an object that has a parent-child relationship with the first object is received before a previous lock request or exclusive access processing on the first object finishes. For simplicity of the description, a case where a lock request is issued from clients 1 to 4 to an

object in layered objects A, B, C in a server 5 as shown in Figure 1 will be considered.

[0023]

A network 6 with the clients 1 to 4 and the server 5 connected thereto forms a local area network (LAN) and the like. Each of the clients 1 to 4 has at least an input device such as a keyboard (not shown) and such output devices as a display, a printer and the like so that a request from a user can be transmitted from the input device to the server 5 via the network 6.

[0024]

The server 5 can store the layered objects (layered objects) A, B, C and the like. When a lock request by an access including a change to any of the layered objects is issued and the object is not locked, the server 5 can generate and store a lock file indicating that the object and objects that have parent-child relationship with the object are locked.

[0025]

It is assumed that a lock request 7-1 by an access including a change in description of an object A-(1) at a place (1) in the layered objects A is issued. Those excluded by the lock request 7-1 include the object A-(1), child objects (I) (RO) of the object A-(1) included in the layers lower than that of the object A-(1) and including the object A-(1) itself, and a parent object A-(2) included in the layer higher than the object A-(1). Exclusion of a lock request by an

access including a change issued from other clients 2 to 4 needs to be performed on the objects.

[0026]

In order to perform such exclusion of a lock request by such an access, generation of lock files under the names to be described below and lock determination are used. An object named D will be described as 'object D' below.

[0027]

When a lock request is issued to the object A-(1) and a lock request occurs to parent objects and child objects of the object A-(1) as mentioned above, a lock file indicated by the lock file name(A) is generated.

[0028]

Unlike the case shown in Figure 1, here, the depth of layers in the layer structure of layered objects A including the object A-(1) is 'm' and the relative depth of layers from the parent object at the highest layer of the object A-(1) to the object A-(1) to which the object request occurs is 'n'.

[0029]

(A) = "P(n).P(m-1)...P(2).P(1).A-(1)..."

(A): lock file name to be generated

A-(1): the name of an object to which a lock request is issued

P(i): the name of parent object that has the object A-(1) as a child object and is present at a layer higher than that of the object A-(1) by i layers

The lock file name (A) to be generated includes the number of periods '..' as many as the depth of layers in the layer structure 'm'. The number of periods '..' included after the name A- (1) is 'm-n', which is the number obtained by subtracting the relative depth from the highest layer object P(n) to the object A- (1) 'n' from the number of layers in the entire layer structure 'm'.

[0030]

Whether the object A-(1) is locked or not is determined as below: When any of the lock files named (A)-(1) to (A)-(n+1) shown below is present, the object A-(1) is determined as locked and when any of the lock files is not present, the object A-(1) is determined as not locked.

[0031]

(A) - (1) : "P(n) P(2) . P(1) . A - (1) . * . * * . "

(A) - (2) : "P(n) P(2) . P(1) "

(A) - (3) : " $P(n), \dots, P(2), \dots, \dots$ "

2

2

(A) - (n+1) : "P(n) , "

Here, '*' included in (A)-(1) means a character string including a null character. Thus, (A)-(1) includes a case where the names of all the child objects in the object A-(1) are described in the place of '*'. Accordingly, it indicates that the object A-(1) and all the child objects of the object A-(1) are locked.

[0032]

The lock file names (A)-(2) to (A)-(n+1) are the lock file names of respective parent objects of the object A-(1). Each of the lock file names (A)-(1) to (A)-(n+1) includes the same number of periods '.', 'm'. (A)-(1) is the lock file name of the object A-(1) or all the child objects of the object A-(1). (A)-(2) to (A)-(n+1) are the lock file names of respective parent objects of the object A-(1).

[0033]

The lock file name (A) may be recorded as a lock file record of the object A-(1). In such a case, the lock file name may be decided according to a predetermined system.

[0034]

In Figure 1, when the lock request 7-1 is issued from the client 1 to the object A-(1) in the server 5 via the network 6 and the object A-(1) is not locked, a lock file of the lock file name (A) is generated and stored to lock the object A-(1), the child objects A-(I) and A-(RO) of the object A-(1) and the parent object A-(2) of the object A-(1). (In such a case, lock files named 'A-(2), A-(1)...' are generated.)

When a lock request 7-2 is issued from the client 2 to an object A-(HA) in the server 5 while the lock request 7-1 is processed, i.e., while the objects are locked by the lock file under the lock file name (A), whether the objects are locked or not is determined based on whether a lock file including the name of the object A-(HA) or any of the names

of the parent objects of the object A-(HA) is present or not (if the lock file is present, it is determined as locked).

[0035]

When the objects A-(2), A-(I), and A-(RO) are locked by the lock file under the lock file name (A) and the lock request 7-2 is issued from the client 2 to the object A-(HA), it is determined that the object A-(HA) is not locked on the condition that only the lock file under the lock file name (A) is present. Then, the lock files named 'A-(2)', 'A-(1)', 'A-(HA)' are generated.

[0036]

When a lock request 7-3 is issued from the client 2 to an object B-(1) next and the object B-(1) is not locked, lock files under the file name (B) such as 'B-(3). B-(2). B-(1)' indicating that the object B-(1) and parent objects B-(3) and B-(2) of the object B-(1) are locked are generated and stored. When another client makes an access including a change to any of the layered objects B while a changing access by the lock request 7-3 is processed, i.e., while the objects are locked by the lock file under the lock file name (B), whether the objects are locked or not is determined based on whether the lock file under the lock file name (B) is present or not.

[0037]

When a lock request 7-4 is to be issued from the client 3 to an object C-(1) and the object C-(1) is not locked,

a lock file under the lock file name (C) of 'C-(1)...' is generated and stored.

[0038]

When another client makes an access including a change to any of the layered objects C while a changing access by the lock request 7-4 is processed, i.e., while the objects are locked by the lock file under the lock file name (C), whether the objects are locked or not is determined based on whether the lock file under the lock file name (C) is present or not.

[0039]

Figure 2 is a diagram illustrating Figure 1 in detail by specifying the layered objects A, B and C and their lock file names. In Figure 2, parent object names at the highest layers of the layered objects A, B and C are described by the names of EDBID1, EDBID2, EDBID3 (EDBID: Engineering Data-Based ID) respectively, object names at the layers lower than those of the layered objects A, B and C are described by Document 1 and Document 2 respectively, and the object names at the lowest layers are described by Sheet 1, Sheet 2 and Sheet 3, respectively.

[0040]

The depth of the layer structure 'm' is '3'. Each of the layered objects includes object sheets 1, 2 and the like and each of the layered objects A, B and C includes an object document 1 and an object document 2. If the lock files are simply named as Sheet 1, Sheet 2 or Document 1, Document

2, the object cannot be identified. Thus, a name indicating all the names traced to the parent object at the highest layer is described as the lock file name.

[0041]

In the description below, the child object A-(1) at a layer immediately lower than that of the object A-(2) included in the object A-(2) is described as 'A-(2)/A-(1)' including the relationship between the layers. When a lock request or a request to obtain lock information is sent from the clients 1 to 4 to the server 5, the object to be locked or the object for which the lock information is to be obtained is reported to the server 5 in the notation.

[0042]

First, in the layered objects A, the object document 2 at the second layer is to be subject to the lock request 7-1. Thus, a lock request is issued to the object of 'EDBID1/Document 2'.

[0043]

As a result, whether the object of 'EDBID1/Document 2' is locked or not is checked by the examination method to be described later. When the object is found to be not locked, a lock file under the lock file name 'EDBID1.Document 2...' is generated according to the abovementioned notation and the objects of 'EDBID1', 'EDBID1/Document 2', 'EDBID1/Document 2/Sheet 1' and 'EDBID1/Document 2/Sheet 2' are locked.

[0044]

When the lock file 'EDBID1. Document 2...' is created in such a manner, the four objects in a dotted line a ('EDBID1', 'EDBID1/Document 2', 'EDBID1/Document 2/Sheet 1', and 'EDBID1/Document 2/Sheet 2') are locked. Whether the objects are locked or not can be determined by a lock state determining method to be described later.

[0045]

Next, in the layered objects B, the object sheet 2 at the lowest layer is to be subject to the lock request 7-3. Thus, a lock request is issued to the object of 'EDBID2/Document 1/Sheet 2'.

[0046]

As a result, whether the object of 'EDBID2/Document 1/Sheet 2' is locked or not is checked by the examination method to be described later. When the object is found to be not locked, a lock file under the lock file name 'EDBID2. Document 1. Sheet 2.' is generated in the abovementioned manner and the objects of 'EDBID2' 'EDBID2/Document 1', and 'EDBID2/Document 1/Sheet 2' are locked.

[0047]

When the lock file is created in such a manner, the three objects in a dotted line b ('EDBID2', 'EDBID2/Document 1', and 'EDBID2/Document 1/Sheet 2') are locked. Whether the objects are locked or not can be determined by a lock state determining method to be described later.

[0048]

Further, in the layered objects C, the object EDBID3 at the highest layer is to be subject to the lock request 7-4. Thus, a lock request is issued to the object of 'EDBID3'.
[0049]

As a result, whether the object of 'EDBID3' is locked or not is checked by the examination method to be described later. When the object is found to be not locked, a lock file under the lock file name 'EDBID3...' is generated in the abovementioned manner and the objects of 'EDBID3', 'EDBID3/Document 1', 'EDBID3/Document 2', 'EDBID3/Document 1/Sheet 1', 'EDBID3/Document 1/Sheet 2' and 'EDBID3/Document 2/Sheet 1' are locked.

[0050]

When the lock file is created in such a manner, the six objects in a dotted line c that are all objects in the layered objects C ('EDBID3', 'EDBID3/Document 1', 'EDBID3/Document 2', 'EDBID3/Document 1/Sheet 1', 'EDBID3/Document 1/Sheet 2', and 'EDBID3/Document 2/Sheet 1') are locked. Whether the objects are locked or not can be determined by a lock state determining method to be described later.

[0051]

Now, examination on whether an object is locked or not will be described. In the case of the layered objects A, the objects in the dotted line a are locked by the lock request 7-1 issued from the client 1. Here, it is assumed that another client 2 or the like issues a request to examine whether

a lock request has been issued to the object of 'EDBID1/Document 2' in the layered objects A or the object of 'EDBID1/Document 2' of the layered objects A is locked (a lock information obtainment request).

[0052]

(1) First, whether a lock file 'EDBID1/Document 2' under the name 'EDBID1.Document 2.*.' or a child object thereof is present or not is examined.

(2) Next, whether a lock file of a parent object of 'EDBID1/Document 2' named 'EDBID1...' is present or not is examined.

[0053]

(3) When no lock file is found at the examinations at (1) and (2), it is determined that the object is not locked. When at least a lock file is found at the examination at (1) or (2), it is determined that the object is locked.

[0054]

In the case of the layered objects B, it is assumed that the objects in the dotted line b are locked by the lock request 7-3 issued from the client 2, and another client 1, 3 or the like issues a lock request or a request to obtain lock information to 'EDBID1/Document 2/Sheet 2' of the layered objects B.

[0055]

In such a case, lock determination is performed on 'EDBID1/Document 2/Sheet 2'.

(1) First, whether a lock file of 'EDBID2/Document1/Sheet 2' itself under the name 'EDBID1. Document 2. Sheet 2.' is present or not is examined.

[0056]

(2) Next, whether a lock file of a parent object of 'EDBID2/Document 1/Sheet 2' named 'EDBID2. Document 1...' or 'EDBID2...' is present or not is examined.

(3) When no lock file is found at the examinations at (1) and (2), it is determined that the object is not locked. When at least a lock file is found at the examination at (1) or (2), it is determined that the object is locked.

[0057]

In the case of the layered objects C, it is assumed that the objects in the dotted line c are locked by the lock request 7-4 issued from the client 3, and another client 1, 2 or the like issues to the object of 'EDBID3' of the layered objects C, a lock information obtainment request of whether a lock request has been issued or the object of 'EDBID3' of the layered objects C is locked.

[0058]

In such a case, lock determination is performed on 'EDBID3'.

(1) Whether a lock file named 'EDBID3.*.*.' is present or not is examined.

(2) When no lock file is found at the examination, it is determined that the object is not locked. When at least

a lock file is found at the examination, it is determined that the object is locked.

[0059]

'*' in the above description means a character string including a null character. Figure 3 is a flowchart of a lock controlling operation performed at the server 5. At step S1, names of objects that are to be subject to the current lock request that is issued from any of the clients via the network 6 (hereinafter referred to as target object) are accepted at the server 5.

[0060]

At step S2, whether the accepted target objects are present in the server 5 or not is examined. At step S3, whether the target objects are present in the server 5 or not is determined based on the result of the examination performed at step S2. When the target objects are present in the server 5, the process proceeds to step S4. When the target objects are not present in the server 5, the process proceeds to step S7.

[0061]

At step S4, examination on whether the target objects are locked or not is performed by using lock determination according to a method for determining whether the lock file described with reference to Figure 2 is present or not on the name of the accepted target object and the names of all the lock files currently present. As a result of the examination, when the target objects are locked, the process

proceeds to step S7, and when the target objects are not locked, the process proceeds to step S5.

[0062]

At step S5, a lock file for performing lock controlling in response to the current lock request is generated, as the examination performed at step S4 shows that no locked object is found in the target objects to be subject to the lock controlling in response to the current lock request.

[0063]

At step S6, lock controlling on the target object that was described with reference to Figure 2 is processed in response to the current lock request. At step S7, whether the target object has been successfully locked or not is reported to the client that issued the current lock request via the network 6.

[0064]

Figure 4 is a flowchart of lock controlling performed by the server 5 from the viewpoint of the client when the lock request accepted at step S1 shown in Figure 3 intends to two or more target objects and a lock request is collectively issued for objects at the third layer that have the same parent object names in the first and the second layer in a three layered objects.

[0065]

Two modes such as ALL mode and POSSIBLE mode are prepared for a collective lock request issued to two or more objects. In the All mode, in order to ensure that all the objects

have been locked, lock on all the objects is cancelled when at least an object cannot be locked. In the POSSIBLE mode, only the objects that can be locked is locked.

[0066]

At step S11, a collective lock request is issued from any of the clients to four objects in the server 5 ('EDBID5/Document 1/Sheet 1', 'EDBID5/Document 1/Sheet 2', 'EDBID5/Document 1/Sheet 3' and 'EDBID5/Document 1/Sheet 5'). The processes performed at steps S1 to step S5 shown in Figure 3 are performed in response to the collective lock request at the server 5.

[0067]

Step S12 shows a case where the processes performed at steps S1 to S5 show that all the target objects are not locked, and accordingly, a lock file is generated for all the target objects and all the target objects are successfully locked.

[0068]

Step S14 shows a case where the collective lock request has been processed in response to the result of step S12 where all the target objects are successfully locked. Step S13 shows a case where only an object ('EDBID5/Document 1/Sheet 3') is unsuccessfully locked as that object has been locked so that a lock file was not generated for that object. In such a case, selection on whether to stop processing to change the objects without locking all the objects, or to lock only the objects that can be locked and change only the objects

is performed according to the lock mode set by the user. In the case shown at step S13, the process proceeds to step S15 or S16 according to the lock mode specified when a collective lock request is issued.

[0069]

Step S15 is the process performed when the user selected the POSSIBLE mode, in which only the objects that are locked in response to the current lock request ('EDBID5/Document 1/Sheet 1', 'EDBID5/Document 1/Sheet 2', 'EDBID5/Document 1/Sheet 5') have been locked except for the object that is unsuccessfully locked 'EDBID5/Document 1/Sheet 3'.

[0070]

Step S16 is the process performed when the user selected the ALL mode, in which lock on the successfully locked target objects ('EDBID5/Document 1/Sheet 1', 'EDBID5/Document 1/Sheet 2', 'EDBID5/Document 1/Sheet 5') is released, i.e., a lock file that has been generated for the target objects is deleted so that the current collective lock is cancelled.

[0071]

As mentioned above, Figure 4 shows that a case where a collective lock request is issued to two or more objects can be addressed: Two modes such as ALL mode and POSSIBLE mode are prepared for a client for ensuring that all the target objects have been locked. In the All mode, whole of the target objects are not locked when at least an object cannot be locked. In the POSSIBLE mode, only the objects that can be locked is locked.

[0072]

Figure 5 is a block diagram of a system in which a controlling server 5 controls lock management in an integrated fashion (generation, deletion of the lock file, lock information obtainment) in the client/server environment. A lock information obtainment request 10 issued from the client 1 and a lock request issued from the client 3 are sent to the controlling server 5. The requests 10 and 11 are collectively processed by the controlling server 5 according to the lock controlling method of the embodiment, and a lock file is generated on the controlling server 5 (in the case of the lock request 11). In this manner, lock management on the layered objects can be performed in an integrated manner on the client/server environment without causing inconsistency among the objects that have parent-child relationship.

[0073]

It is assumed that an object A is desired to be updated by using an editor or the like at the client 1. In such a case, the lock information obtainment request 10 is issued to the controlling server 5 to obtain lock information on the object.

[0074]

When the controlling server 5 receives the lock information obtainment request 10, it determines whether the object A is locked or not. The determination is performed by checking whether the lock file of the object A and objects

that have parent-child relationship with the object A is present or not. The controlling server 5 returns the determination to the client 1.

[0075]

If the returned determination tells that the object A is locked, the client 1 can access the object A only in a reading mode. If the returned determination tells that the object A is not locked, the client 1 can access the object A in an updating mode as well as the reading mode.

[0076]

When the client 1 accesses the object A in the updating mode, the object A needs to be locked so that the object A is not updated by another access during the update performed by the client 1. For that purpose, the client 1 needs to send a lock request to the controlling server 5.

[0077]

In response to the lock request, the controlling server 5 generates a lock file of the object A on the controlling server 5. After the lock file of the object A is generated, the object A can be freely updated at the client 1.

[0078]

When no object needs to be locked after the object A has been updated by the client 1, the client 1 sends a lock release request for the object A to the controlling server 5. In response to the lock release request, the controlling server 5 deletes the lock file of the object A.

[0079]

The procedure shown above includes the steps of sending the lock information obtainment request 10 for the object A to the controlling server 5 by the client 1 and receiving the determination obtained at the controlling server 5 by the client 1 at first, and sending the lock request by the client 1 on the condition that the object A is not locked. The first step of sending the lock information obtainment request 10 and receiving the determination may be omitted and the lock request 11 may be sent at first as the lock request 11 is sent from the client 3 as shown in Figure 5.

[0080]

The reason that the lock request can be sent first is that, the controlling server 5 determines whether the object to which the lock request is issued is locked or not each time it receives the lock request 11, decides whether or not to lock the object to which the lock request is issued based on the determination, and returns the information on whether the object has been successfully locked or not to the client side.

[0081]

Figure 6 shows a case where a collective lock request is sent to the controlling server 5 in the form of packet information 9 to reduce the traffic load on the network when a collective lock request is issued for the abovementioned objects at the clients 1 to 4.

[0082]

In the case shown in Figure 6, a piece of packet information 9 includes a lock request command 9-0 indicating that the packet information is information on the lock request to the server 5, lock mode information 9-1 in which a lock mode selected from the POSSIBLE mode and the ALL mode by the user is set, delimiters 9-2, 9-4, 9-6, 9-8, 9-9, and name information 9-3, 9-5, 9-7 on objects at respective layers each of which is inserted in the space separated by the delimiters.

[0083]

Target objects at the third layer in the three layered objects to which a lock request is issued ('EDBID/Document ID/Sheet 1', 'EDBID/Document ID/Sheet 2', ..., 'EDBID/Document ID/Sheet n') are made common in the first layer and the second layer of the packet information 9 and sent in the form of 'lock mode: \$n: EDBID: \$n: document ID: \$n: sheet 1, sheet 2, ..., sheet n: \$n: \$n:'.

[0084]

If only one of the first objects to which the collective lock request shown in Figure 4 is issued 'EDBID5/Document 1/Sheet 1' (an object at the third layer in the three layer structure) is included in the packet, it is described as 'lock mode: \$n : EDBID5: \$n : Document 1: \$n : sheet 1'. Packets need to be created for the sheets 2, 3 and 5 of the other objects respectively and serially sent out to the controlling server 5. When the four objects to which a collective lock request is issued at step S11 shown in Figure 4 are collectively sent, they are described as 'lock mode:

¥n : EDBID5: ¥n : document 1: ¥n: sheet 1, sheet 2, sheet 3, sheet 5', as they are the same up to the 'EDBID5/document 1'. Thus, only one packet needs to be sent to the controlling server 5. That can reduce the traffic load on the network 6.

[0085]

The sent packet information 9 is converted into such forms as 'EDBID/document ID/sheet 1', 'EDBID/document ID/sheet 2', ...'EDBID/document ID/sheet n' each of which indicates each object at the controlling server 5, subject to lock controlling for each object. When lock files are to be generated, the abovementioned forms that indicate the objects are further converted into forms that indicate the lock file names of the objects, and the lock files under those names are generated.

[0086]

When each client collectively sends information on objects to which a lock request is issued to the server 5 in the form of packet information 9 in the abovementioned manner, the traffic load on the network 6 can be reduced.

[0087]

Figure 7 is a block diagram of an embodiment in which the program for implementing the lock controlling method on layered objects in the information processing system environment according to the embodiment is stored in the storage.

[0088]

An information processing device 21, which corresponds to the server 5 shown in Figure 1 and Figure 5, is a device that can perform information processing such as a mainframe, a workstation or a personal computer. The information processing device 21 includes a storage 24 that can store a program (data) 26 for causing the lock controlling according to the embodiment to be performed, a display 23 that functions as an output device that can output what is stored, a processing result and the like, a medium driving device 25 for driving a transportable storage medium 31, and an input device, a processing device and the like (not shown).

[0089]

A program (data) provider 41 is connected with the information processing device 21 via a circuit 51. The program (data) provider 41 can transfer a program (data) 42 stored thereinto to the storage 24 of the information processing device 21 via the circuit 51 and store the program (data) 42 as the program (data) 26.

[0090]

The transportable storage medium 31 stores a program or data that is processed in the information processing device 21 such as a CD-ROM 32 that stores a program (data) 34 or a floppy disc 33 that stores a program (data) 35, for example. When the transportable storage medium 31 is inserted into the medium driving device 25 of the information processing device 21 and the programs (data) 34, 35 are read by the medium driving device 25, the programs (data) 34, 35 can

be stored in the storage 24 of the information processing device 21 as the program (data) 26.

[0091]

Whether a program (data) is originally the program (data) 42 that is sent from the program (data) provider 41 or the program (data) that is stored as the program (data) 34 from the CD-ROM 32 or the program (data) 35 from the floppy disc 33 in the transportable storage medium 31, it is finally executed as the program (data) 26 stored in the storage 24 of the information processing device 21. Thus, the program (data) can be equally treated without limited by the medium that stores the program (data).

[0092]

[Advantage of the Invention]

According to the present invention, only a first object to which a lock request is issued and a second object that has parent-child relationship with the first object are locked among layered objects. Therefore, any access request to an object that has no parent-child relationship with the first object is enabled thereafter. That enables effective access exclusion (lock management) that does not lower access efficiency to the objects.

[0093]

Further, only a parent object and a child object that are required to be locked are locked. Thus, when another lock request occurs to an object that has parent-child relationship with an object that is processed by a lock request,

logical and simple lock controlling (exclusive controlling) is available without leading any inconsistency to the relationship between the objects.

[0094]

Also, when either parent object or child object of an object has been locked among objects to which a collective lock request is issued, whether to cancel lock on all the objects to which the lock request is issued or to lock only unlocked objects can be selected. That makes the lock controlling flexible.

[0095]

As a server collectively controls lock controlling on layered objects in a client/server environment, simple lock controlling is performed in the client/server environment without leading any inconsistency due to a change among the objects. As lock requests are collectively received from a client to objects in the form of packets, a transmission load in the network can be reduced.

[Brief Description of the Drawings]

[Figure 1]

Figure 1 is a diagram showing the entire configuration of a client/server system according to an embodiment of the present invention.

[Figure 2]

Figure 2 is a diagram illustrating a lock controlling method of layered objects according to the embodiment.

[Figure 3]

Figure 3 is a flowchart illustrating lock controlling performed at a server.

[Figure 4]

Figure 4 is a flowchart illustrating lock controlling performed in ALL mode and POSSIBLE mode when a collective lock request is issued for objects.

[Figure 5]

Figure 5 is a diagram showing a method for collectively managing lock information performed by a server in the client/server system.

[Figure 6]

Figure 6 is a diagram showing an example of packet information form taken in a collective lock request to the server.

[Figure 7]

Figure 7 is a diagram illustrating the configuration of a program for the embodiment of the present invention stored in a storage to be executed.

[Description of Symbols]

1 to 4 client

5 server

6 network

7-1 to 7-4 lock request

9 packet information

10 lock information obtainment request

11 lock request

A, B, C layered objects

(A), (B), (C) lock file name

Figure 1

#1 FIGURE SHOWING ENTIRE CONFIGURATION
#2 LAYERED OBJECTS
#3 (I X RO)
#4 (HA)
#5 LOCK FILE
1-4 CLIENT
5 SERVER
6 NETWORK
7-1...7-4 LOCK REQUEST

Figure 2

#1 FIGURE ILLUSTRATING LOCK CONTROLLING METHOD OF LAYERED
OBJECTS ACCORDING TO THE EMBODIMENT
#2 LAYERED OBJECTS
#3 LOCK FILE NAME
#4 DOTTED LINE a
#5 DOCUMENT 1
#6 A- (HA)
#7 SHEET 1
SHEET 2
SHEET 3
SHEET 1
SHEET 2
#8 DOCUMENT 2
#9 A- (I)
#10 A- (RO)

```
#11 LOCK FILE NAME: EDBID1. DOCUMENT 2..  
#12 DOTTED LINE b  
#13 DOCUMENT 1  
#14 DOCUMENT 2  
#15 SHEET 1  
    SHEET 2  
    SHEET 1  
#16 LOCK FILE NAME: EDBID 2. DOCUMENT 1: SHEET 2.  
#17 DOTTED LINE c  
#18 DOCUMENT 1  
#19 DOCUMENT 2  
#20 SHEET 1  
    SHEET 2  
    SHEET 1  
#21 LOCK FILE NAME: EDBID 3...  
#22 OBJECT TO WHICH LOCK REQUEST IS ISSUED  
#23 OBJECT BEING LOCKED
```

Figure 3

```
#1 FLOWCHART OF LOCK CONTROLLING AT SERVER  
#2 START  
#3 END  
S1 ACCEPT OBJECT NAME  
S2 SEARCH FOR OBJECT NAME  
S3 DOES OBJECT NAME EXIST?  
S4 DOES LOCK FILE EXIST?  
S5 GENERATE LOCK FILE
```

S6 LOCK CONTROLLING PROCESSING
S7 REPORT WHETHER LOCK REQUEST IS SUCCESSFULLY DONE OR
NOT

Figure 4

#1 FLOWCHART OF COLLECTIVE LOCKING OF OBJECTS
#2 POSSIBLE MODE
#3 ALL MODE
#4 EDBID 5/DOCUMENT 1/SHEET 1
EDBID 5/DOCUMENT 1/SHEET 2
EDBID 5/DOCUMENT 1/SHEET 3
EDBID 5/DOCUMENT 1/SHEET 5
S11 COLLECTIVE LOCK REQUEST
S12 SUCCEEDED IN LOCKING ALL
S13 FAILED IN LOCKING ONLY EDBID 5/DOCUMENT 1/SHEET 3
S14 COLLECTIVE LOCKING COMPLETED
S15 LOCKING COMPLETED IN ONLY EDBID 5/DOCUMENT 1/SHEET 1,
EDBID 5/DOCUMENT 1/SHEET 2, EDBID 5/DOCUMENT 1/SHEET 5
S16 RELEASE LOCKING ON EDBID 5/DOCUMENT 1/SHEET 1, EDBID
5/DOCUMENT 1/SHEET 2, EDBID 5/DOCUMENT 1/SHEET 5 AND CANCEL
COLLECTIVE LOCKING

Figure 5

#1 FIGURE SHOWING COLLECTIVE MANAGEMENT ON LOCK INFORMATION
BY SERVER
#2 LOCK FILE GROUP (A), (B), (C)
#3 OBJECTS A, B, C

1-3 CLIENT
5 CONTROLLING SERVER
6 NETWORK
10 LOCK INFORMATION OBTAINMENT REQUEST
11 LOCK REQUEST

Figure 6

#1 FIGURE SHOWING PACKET INFORMATION COLLECTIVELY LOCKED TO SERVER

#2 COLLECTIVE LOCK REQUEST OCCUR

#3 CLIENT 1 OR 2 OR 3... . SIDE

#4 SERVER 5 SIDE

9 PACKET INFORMATION

9-0 LOCK REQUEST COMMAND

9-1 LOCK MODE

9-5 DOCUMENT ID

9-7 SHEET 1, SHEET 2, . . . , SHEET n

Figure 7

#1 INSERT
21 INFORMATION PROCESSING DEVICE
23 DISPLAY
24 STORAGE SUCH AS RAM/ROM HARD DISK
25 MEDIUM DRIVING DEVICE
26 PROGRAM (DATA)
31 TRANSPORTABLE STORAGE MEDIUM
33 FLOPPY DISC

34, 35 PROGRAM (DATA)
41 PROGRAM (DATA) PROVIDER
42 PROGRAM (DATA)
51 CIRCUIT